Adaptation de domaine parcimonieuse par ponderation de bonnes fonctions de similarité

Emilie Morvant, Stéphane Ayache, Amaury Habrard

Adaptation de domaine parcimonieuse par pondération de bonnes fonctions de similarité

Émilie Morvant, Stéphane Ayache, Amaury Habrard

Laboratoire d'Informatique Fondamentale de Marseille
UMR CNRS 6166, Aix-Marseille Université
39 rue F. Joliot-Curie, 13453 Marseille cedex 13
prenom.nom@lif.univ-mrs.fr

Résumé : L'adaptation de domaine est une problématique importante dans laquelle les données *sources* d'apprentissage et les données *cibles* de test sont supposées suivre deux distributions différentes. Nous nous plaçons dans le cadre difficile où aucune information sur les étiquettes cibles n'est disponible. D'un point de vue théorique, Ben-David *et al.* ont montré qu'un classifieur a de meilleures garanties de généralisation lorsque les distributions marginales des données sources et cibles sont proches. Nous présentons une approche basée sur un cadre de Balcan *et al.* permettant l'apprentissage de classifieurs linéaires à partir de fonctions de similarité n'ayant besoin ni d'être symétriques ni d'être semi-définies positives. Nous exploitons cette propriété pour repondérer la fonction de similarité afin de construire itérativement un espace de projection dans lequel les deux distributions marginales sont proches. Notre approche, formulée sous la forme d'un programme linéaire en norme 1, infère des modèles très parcimonieux montrant de bonnes performances d'adaptation. Nous l'évaluons expérimentalement sur des données synthétiques et des corpus réels d'annotations d'images. **Mots-clés** : Apprentissage Multi-Tâches et Transfert, Méthodes à Noyaux, Adaptation de Domaine.

1. Introduction

Pour de nombreux problèmes d'apprentissage automatique, il est généralement admis que les données d'apprentissage sont représentatives des données test. Cependant, cette hypothèse forte n'est pas toujours vérifiée en pratique. Cet inconvénient peut être contourné par des méthodes de *Transfer Learning* (Pan & Yang, 2009) où le but est d'adapter un modèle depuis un domaine source vers un domaine cible. Ce papier s'intéresse au problème d'adaptation de domaine (AD) pour lequel les données test sont supposées tirées selon une distribution - le *domaine cible* - différente de celle génératrice des données

d'apprentissage - le *domaine source* (Quionero-Candela *et al.*, 2009). Différentes approches d'AD ont été proposées dans la littérature pour améliorer les performances de certaines méthodes classiques. Alors qu'une majorité d'entre elles supposent l'utilisation d'étiquettes cibles (Ben-David *et al.*, 2010a; Bergamo & Torresani, 2010; Daumé III, 2007; Schweikert *et al.*, 2008), nous considérons le cadre plus difficile où aucune de ces étiquettes n'est disponible. Ben-David *et al.* (2010a); Mansour *et al.* (2009) ont démontré qu'un classifieur appris uniquement sur des données sources étiquetées peut être performant sur les données cibles si les distributions maginales source et cible sont proches, sous l'hypothèse que les domaines soient corrélés. Intuitivement, une AD judicieuse équivaut à rapprocher les distributions tout en maintenant de bonnes performances sur le domaine source. De cette idée découle différentes méthodes de repondération des données sources en fonction de différentes hypothèses ou mesures de divergence (Huang *et al.*, 2006; Mansour *et al.*, 2009; Sugiyama *et al.*, 2007). Bruzzone & Marconcini (2010) ont quant à eux proposé un processus itératif basé sur les SVMs : à chaque étape, des exemples sources sont retirés de l'échantillon d'apprentissage et des exemples cibles étiquetés par le modèle courant sont ajoutés. Une autre approche vise à construire un nouvel espace de projection où les deux distributions sont proches (Ben-David *et al.*, 2006). Cependant, ce type de méthodes repose essentiellement sur des heuristiques et reste spécifique à certaines tâches. Dans cet article, nous proposons une méthode d'AD pour la classification binaire, inspirée du cadre récent de Balcan *et al.* (2008a,b) autorisant l'apprentissage à partir d'une *bonne fonction de similarité*, n'ayant besoin d'être ni symétrique ni semi-définie positive (SDP) (i.e. qui n'est pas un noyau). Ils ont montré qu'un classifieur linéaire performant peut être appris dans un espace de projection défini par un ensemble de similarités vis-à-vis de points dits *raisonnables*. Notre idée est de modifier automatiquement cet espace, en rapprochant les points sources et cibles, pour aboutir à une bonne adaptation. Pour ce faire, nous proposons une régularisation centrée sur les points raisonnables à la fois proches des exemples sources et cibles. Notre approche est itérative est exploite les propriétés d'une fonction de similarité en la repondérant pour construire un nouvel espace de projection dans lequel les deux distributions sont proches et tout en la maintenant *bonne*. Cette méthode offre ainsi une meilleure flexibilité que celles basées sur les SVMs. Sa formulation en norme 1 permet d'obtenir des modèles très parcimonieux. Nous l'évaluons sur un problème jouet, ainsi que sur un corpus réel d'annotation d'images.

Ce papier est organisé comme suit. Les Sections 2 et 3 introduisent res-

pectivement les cadres d'AD et de Balcan *et al.* (2008a). Notre méthode est présentée en Section 4, puis est évaluée expérimentalement en Section 5.

2. Adaptation de Domaine

Soient X l'espace d'entrée et $Y = \{-1, 1\}$ l'ensemble d'étiquettes. Un domaine est défini comme une distribution de probabilité selon $X \times Y$. Dans le cadre de l'AD, les distributions différentes P_S et P_T désignent respectivement le *domaine source* et le *domaine cible*. D_S et D_T sont les distributions maginales respectives selon X. Un algorithme d'AD prend en entrée un *échantillon source étiqueté*, $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, *i.i.d.* selon P_S et à un *échantillon cible non-étiqueté*, $TS = \{\mathbf{x}_j\}_{j=1}^{m'}$, *i.i.d.* selon D_T.
Soit $h : X \to Y$ une hypothèse. L'espérance de l'erreur de h sur le domaine source P_S est la probabilité qu'elle fasse une erreur de classification :

$$err_S(h) \;=\; \mathbb{E}_{(\mathbf{x},y)\sim P_S}\, L_{01}\big(h(\mathbf{x}) - y\big)$$

où $L_{01}(h(\mathbf{x}) - y)$ est la *fonction perte* 0-1 et vaut 1 si $h(\mathbf{x}) \neq y$ et 0 sinon. L'erreur err_T sur le domaine cible est définie de manière équivalente. $\hat{err}_S$ et $\hat{err}_T$ sont les erreurs empiriques associées. On note $\mathcal{H}$ la classe d'hypothèses de X vers Y considérée.

Nous présentons maintenant le cadre théorique d'AD de Ben-David *et al.* (2006, 2010a) définissant une borne majorant l'erreur sur le domaine cible.

Théorème 1 (Ben-David *et al.* (2006, 2010a))
Soit $\mathcal{H}$ une classe d'hypothèses symétrique, pour tout $h \in \mathcal{H}$:

$$err_T(h) \;\leq\; err_S(h) + \frac{1}{2} d_{\mathcal{H}}(D_S, D_T) + \nu,$$

où $\nu = err_S(h^*) + err_T(h^*)$ avec $h^* = \operatorname{argmin}_{h\in\mathcal{H}}\, err_S(h) + err_T(h)$.

Cette borne dépend de l'erreur sur le domaine source qui peut facilement se minimiser selon le principe ERM. Le terme ν correspond à l'erreur jointe de la meilleure hypothèse possible sur les deux domaines. Plus elle est grande, plus inférer un modèle performant sur le domaine cible sera dur. Elle peut donc être considérée comme mesure de qualité de $\mathcal{H}$ pour le problème considéré.
L'autre point important est la divergence[1] $d_{\mathcal{H}}$. Le Th. 1 suggère que si les distributions sont proches, alors un classifieur d'erreur faible sur le domaine

[1] Les auteurs considèrent la divergence sur $\mathcal{H}\Delta\mathcal{H}$, l'espace de la différence symétrique des hypothèses. Voir Ben-David *et al.* (2010a) pour plus de détails.

source peut être performant sur le domaine cible. En fait, cette mesure, liée à $\mathcal{H}$ et appelée la $\mathcal{H}$-distance, calcule la variation maximale entre les points pour lesquels une hypothèse de $\mathcal{H}$ peut commettre une erreur : $d_{\mathcal{H}}(D_S, D_T) = 2\sup_{h\in\mathcal{H}} \left|P_{D_S}[I(h)] - P_{D_T}[I(h)]\right|$ où $\mathbf{x} \in I(h) \Leftrightarrow h(\mathbf{x}) = 1$. Si $\mathcal{H}$ admet une dimension VC finie, alors $d_{\mathcal{H}}$ est estimable sur des échantillons finis de sorte que la divergence empirique $\hat{d}_{\mathcal{H}}$ converge vers $d_{\mathcal{H}}$ avec la taille des échantillons. Si nous considérons l'échantillon $U \cup U'$ étiqueté tel que chaque instance de U soit positive et chaque instance de U' négative, alors la divergence empirique est estimable à partir de l'hypothèse optimale capable de séparer les deux échantillons Ben-David *et al.* (2010a) :

$$\hat{d}_{\mathcal{H}}(U, U') \;=\; 2\big(1 - \min_{h\in\mathcal{H}} \underset{U,U'}{\hat{err}}(h)\big) \tag{1}$$

avec $\underset{U,U'}{\hat{err}}(h) \;=\; \frac{1}{m}\Big(\sum_{\mathbf{x}:h(\mathbf{x})=-1} \mathbb{1}_{\mathbf{x}\in U} + \sum_{\mathbf{x}:h(\mathbf{x})=1} \mathbb{1}_{\mathbf{x}\in U'}\Big)$, où $\mathbb{1}_{\mathbf{x}\in U} = \begin{cases} 1 \text{ si } \mathbf{x}\in U \\ 0 \text{ sinon.} \end{cases}$

En général, inférer cet hyperplan optimal est NP-dur. Cependant, une bonne estimation de $\hat{d}_{\mathcal{H}}$ permet d'avoir un aperçu de la distance entre les deux distributions marginales et donc de la difficulté du problème d'AD pour la classe $\mathcal{H}$. Notons que Mansour *et al.* (2009) ont étendu la $\mathcal{H}$-distance à une fonction à valeurs réelles et ont montré des bornes de Rademacher en généralisation.

Le Th.1 suggère qu'un bon algorithme d'AD doit inférer un espace de projection tel que la $\mathcal{H}$-distance et l'erreur du classifieur sur le domaine source soient faible. Selon Ben-David *et al.* (2010b), minimiser ces deux termes apparaît nécessaire pour assurer d'une bonne adaptation.

3. Apprentissage à partir d'une Bonne Fonction de Similarité

Cette section présente la classe $\mathcal{H}$ de classifieurs linéaires définis à partir d'une similarité. Toute fonction $K : X \times X \to [-1, 1]$ peut être considérée comme fonction de similarité sur X. De nombreux algorithmes font appel à de telles fonctions. C'est le cas des SVMs qui utilisent une similarité symétrique et semi-définie positive (SDP) appelée noyau. L'apprentissage s'effectue alors implicitement dans l'espace de grande dimension définit par ce noyau. Cependant être SDP est une contrainte forte et la définition d'un bon noyau reste une tâche difficile en général. Afin de contourner certaines de ces limitations, Balcan *et al.* (2008a,b) ont récemment proposé un cadre d'apprentissage où la définition d'une bonne fonction de similarité est plus intuitive. Nous allons le présenter en commençant par cette définition.

Définition 1 (Balcan *et al.* (2008a))
*Une fonction de similarité K est une (ϵ, γ, τ)-***bonne fonction de similarité** *pour un problème P s'il existe un fonction indicatrice (aléatoire) $R(\mathbf{x})$ qui définit un ensemble de points raisonnables tel que les conditions suivantes soient vérifiées :*

(i) une proportion $1 - \epsilon$ des exemples $(\mathbf{x}, y)$ satisfont

$$\mathbb{E}_{(\mathbf{x}',y')\sim P}\big[yy'K(\mathbf{x},\mathbf{x}')|R(\mathbf{x}')\big] \;\geq\; \gamma,$$

(ii) $Pr_{\mathbf{x}'}[R(\mathbf{x}')] \geq \tau$.

En d'autres termes, la majorité des exemples doivent être plus similaires aux points raisonnables de même classe qu'à ceux de classe opposée, avec une confiance γ. De plus, au moins une proportion τ des points doivent être raisonnables. La Def.1 inclut aussi bien des noyaux valides que des similarités non-SDP non-symétrique : c'est une généralisation des noyaux (Balcan *et al.*, 2008a,b). Le Th. 2 pose les conditions suffisantes pour apprendre un classifieur linéaire performant dans l'espace induit par les points raisonnables.

Théorème 2 (Balcan *et al.* (2008a))
Soit K une (ϵ, γ, τ)-bonne fonction de similarité pour un problème d'apprentissage P. Soit $S = \{x'_1, .., x'_d\}$ un échantillon de $d = \frac{2}{\tau}(\log\frac{2}{\delta} + 8\frac{\log(2/\delta)}{\gamma^2})$ points raisonnables (non-étiquetés) tirés selon P. Considérons $\phi^R : X \to \mathbb{R}^d$ la projection définie par $\phi^R_i(x) = K(x, x'_i),\ i \in \{1, .., d\}$ induite par les similarités avec les points de R. Alors, avec une probabilité d'au moins $1 - \delta$ sur l'échantillon aléatoire R, la distribution induite $\phi^R(P)$ dans $\mathbb{R}^d$ admet un séparateur d'erreur au plus $\epsilon + \delta$ par rapport à une marge L_1 d'au moins $\gamma/2$.

Pour un problème donné, si K est une (ϵ, γ, τ)-bonne fonction de similarité associée à une quantité pertinente de points raisonnables, alors il existe avec une grande probabilité un séparateur linéaire d'erreur faible dans l'espace ϕ^R (espace induit par les similarités avec les d points raisonnables). Ainsi, avec du exemples non-étiquetés et dl exemples étiquetés, un séparateur défini par $\boldsymbol{\alpha} \in \mathbb{R}^{du}$ peut être inféré efficacement en résolvant un problème d'optimisation linéaire. Ceci conduit à un algorithme d'apprentissage en deux étapes : sélectionner un ensemble de points raisonnables puis apprendre un classifieur linéaire dans l'espace de projection induit. L'apprentissage de $\boldsymbol{\alpha}$ est basée sur la perte Hinge ($[1 - z]_+ = \max(0, 1 - z)$) (Balcan *et al.*, 2008a) :

$$\min_{\boldsymbol{\alpha}} \;\sum_{i=1}^{dl}\left[1 - \sum_{j=1}^{du}\alpha_j y_i K(x_i, x'_j)\right]_+, \qquad \text{s.t.} \sum_{j=1}^{du}|\alpha_j| \;\leq\; 1/\gamma \qquad (2)$$

Nous dénotons par "classifieur SF" un classifieur linéaire appris via ce principe. Une formulation en programmation linéaire du Pb.(2) sera proposée.

4. Adaptation de Domaine par Pondération d'une Fonction de Similarité

Nous présentons maintenant notre méthode d'AD basée sur la pondération d'une bonne fonction de similarité. Rappelons que dans le cadre de Balcan *et al.*, l'apprentissage d'un classifieur linéaire performant est possible dans l'espace ϕ^R des similarités à l'ensemble des points raisonnables. D'après le Th.1, nous cherchons un espace ϕ^R tel que les distributions source et cible soient proches. Pour ce faire, nous proposons une régularisation aidant à la selection des points raisonnables pertinents pour les deux domaines. La repondérantion des similarités, basée sur le classifieur linéaire appris, induit un nouvel espace ϕ^R. Le processus est ensuite itéré jusqu'à un critère d'arrêt défini à l'aide d'une approche par *validation inverse*.

4.1. La Méthode d'Apprentissage Proposée

Considérons dl points sources étiquetés, $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^{dl}$, tirés selon P_S, du points raisonnables, $R = \{\mathbf{x}'_j\}_{j=1}^{du}$, tirés selon D_S et un échantillon d'exemples cibles non-étiquetés TS tirés selon D_T.

Tout d'abord, nous reformulons le Pb.(2) comme un programme linéaire classique en y introduisant des *slack* variables $\boldsymbol{\xi}$ gérant la perte Hinge et en approximant la contrainte $L1$ par une régularisation en norme 1 pondérée par un paramètre λ. Cette régularisation est reformulée en remplaçant les variables $\boldsymbol{\alpha}$ par des ensembles de variables $\boldsymbol{\alpha}^+$ et $\boldsymbol{\alpha}^-$, tels que $\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-$. Elle n'est pas équivalente à la contrainte $L1$ mais tend à la satisfaire.

Définissons maintenant notre terme de régularisation. Nous selectionnons deux sous-ensemble U_S et U_T de même taille issus respectivement de LS et TS. Ensuite, nous construisons un couplage biparti, $\mathcal{C}_{ST}$, entre les points de U_S et U_T en minimisant : $\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}} \left\|\phi^R(\mathbf{x}_s) - \phi^R(\mathbf{x}_t)\right\|_2^2$. Étant donné cet ensemble $\mathcal{C}_{ST}$ qui contient les paires d'exemples *(source,cible)* de $U_S \times U_T$ proches, nous voulons selectionner les points raisonnables - donc $\boldsymbol{\alpha}$ - qui minimisent la norme 1 : $\|({}^t\phi^R(\mathbf{x}_s) - {}^t\phi^R(\mathbf{x}_t))\,\mathrm{diag}(\boldsymbol{\alpha})\|_1$ pour chaque paire $(\mathbf{x}_s, \mathbf{x}_t) \in \mathcal{C}_{ST}$ (où ^{t} défini la transposée). Grâce à la présence de $\boldsymbol{\alpha}$, ce terme tend à selectionner un espace dans lequel des points sources et cibles sont difficiles à séparer, ce qui, d'après l'Eq.(1), tend à une décroissance de $\hat{d}_{\mathcal{H}}$. Notons que ces choix sont effectués arbitrairement puisque la distribution cible

est inconnue. Nous intégrons cette régularisation dans un programme linéaire en introduisant des *slack* variables $\boldsymbol{\eta}$, coefficientées par un paramètre C.

Notre problème d'optimisation global est défini dans l'Eq.(3) : un programme linéaire classique et efficacement résoluble en temps polynomial. La régularisation en norme 1 infère naturellement des modèles parcimonieux.

$$\begin{aligned}
\min_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-,\boldsymbol{\xi},\boldsymbol{\eta}} \sum_{i=1}^{dl}\xi_i + \sum_{j=1}^{du}\left(C\sum_{(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}}\eta_{st}^j + \lambda\left(\alpha_j^+ + \alpha_j^-\right)\right) \\
\text{s.t. } \forall i = 1,\dots,dl: \quad \xi_i \;\geq\; 0, \quad \xi_i \;\geq\; 1 - \sum_{i=1}^{dl}(\alpha_j^+ - \alpha_j^-)y_i K(\mathbf{x}_i,\mathbf{x}'_j) \\
\forall j = 1,\dots,du, \forall(\mathbf{x}_s,\mathbf{x}_t)\in\mathcal{C}_{ST}: \quad \alpha_j^+ \;\geq\; 0,\; \alpha_j^- \;\geq\; 0 \\
\eta_{st}^j \;\geq\; (\alpha_j^+ - \alpha_j^-)(K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)) \\
\eta_{st}^j \;\geq -(\alpha_j^+ - \alpha_j^-)(K(\mathbf{x}_s,\mathbf{x}'_j) - K(\mathbf{x}_t,\mathbf{x}'_j)). \quad (3)
\end{aligned}$$

Les points raisonnables sélectionnés par l'algorithme, *i.e.* ceux associés à un poids α_j non nul, définissent une projection dans laquelle les deux distributions sont plus proches. En effet, ils contribuent à la minimisation de la norme 1 sur les $\mathcal{C}_{ST}$, ce que nous exploitons ensuite en proposant une approche itérative repondérant la fonction de similarité les $\boldsymbol{\alpha}$. Ce processus peut être vu comme une contraction de l'espace pour minimiser $\hat{d}_{\mathcal{H}}$. Supposons qu'à l'itération l et avec la fonction de similarité K_l (repondérée), la résolution de l'Eq.(3) nous renvoie de nouveaux poids $\boldsymbol{\alpha}^l$. Nous proposons alors de définir K_{l+1} en pondérant K_l vis-à-vis de chacun des points raisonnables $\mathbf{x}'_j \in R$ (en la renormalisant éventuellement pour assurer $K_{l+1} \in [-1,1]$) telle que :

$$K_{l+1}(\mathbf{x},\mathbf{x}'_j) \;=\; \alpha_j^l K_l(\mathbf{x},\mathbf{x}'_j).$$

Une bonne fonction de similarité n'a pas besoin d'être symétrique ou SDP. Notre normalisation est valide, si et seulement si, la nouvelle similarité est encore suffisamment bonne sur le domaine source (notons que l'ensemble R est mis à jour pour ne contenir que les points de poids non nul). Ce processus de repondération permet en fait de relier la qualité de la similarité au classifieur h_l appris à l'itération l. En effet, pour tout x, on a :

$$\mathbb{E}_{\mathbf{x}'\in R}\left[K_{l+1}(\mathbf{x},\mathbf{x}')\right] = \tfrac{1}{du}\sum_{j=1}^{du}K_{l+1}(\mathbf{x},\mathbf{x}'_j) = \tfrac{1}{du}\sum_{j=1}^{du}\alpha_j^l K_l(\mathbf{x},\mathbf{x}'_j) = \tfrac{1}{du}h_l(x).$$

Cette qualité dépend donc de celle du classifieur, contrôlée par le terme issu de l'Eq.(3) de minimisation de l'erreur sur le domaine source ($\sum_i \xi_i$).

4.2. Classifieur Inverse et Validation

Étant donné un classifieur h_l, son *classifieur inverse* h_l^r est le classifieur appris dans l'espace ϕ_l^R (considéré à l'itération l) à partir des exemples cibles étiquetés par h_l : $\{(\mathbf{x}, \text{sign}(h_l(\mathbf{x})))\}_{\mathbf{x}\in TS}$. En s'inspirant de l'idée de Zhong *et al.* (2010), nous évaluons la qualité de notre classifieur h_l à l'aide de celle du classifieur inverse sur le domaine source. Une idée similaire a été proposée en tant que validation circulaire par Bruzzone & Marconcini (2010). Étant donnés k sous-ensembles de l'échantillon étiqueté source ($LS = \cup_{i=1}^k LS_i$), nous en utilisons k-1 en tant qu'exemples étiquetés pour résoudre le Pb.(3) et évaluons h_l^r sur le dernier sous-ensemble. L'erreur finale correspond à la moyenne des erreurs sur les k sous-ensembles : $\hat{err}_S(h_l^r) = \frac{1}{k}\sum_{i=1}^k \hat{err}_{LS_i}(h_l^r)$.

4.3. Critère d'Arrêt

D'après le Th.1, l'erreur sur le domaine cible est bornée par la somme de trois termes : (i) l'erreur sur le domaine source, (ii) la distance entre les deux distributions et (iii) l'erreur jointe du meilleur classifieur sur les deux domaines. Résoudre le Pb.(3) produit une décroissance naturelle de (i) et (ii). L'erreur jointe (iii) peut être associée à la capacité d'adaptation de notre classe d'hypothèses $\mathcal{H}$ dans l'espace de projection courant. Contrôler ce terme et sa diminution au cours des itérations semblerait être un critère attractif pour l'arrêt de l'agorithme. Cependant, puisqu'aucune information sur le domaine cible n'est disponible, nous proposons d'évaluer l'erreur jointe, à une itération l donnée, comme étant l'erreur du classifieur inverse h_l^r sur les deux domaines. La justification de ce choix est liée au fait que si les domaines sont suffisamment proches et reliés, alors le classifieur inverse doit être performant sur la tâche source (Bruzzone & Marconcini, 2010). En d'autres termes, nous devons être capable de passer d'un problème à l'autre dans l'espace de projection. Si l'erreur jointe, estimée par h_l^r, augmente entre deux itérations, le nouvel espace construit n'est plus pertinent et l'espace précédent est à préférer. L'erreur de h_l^r sur le domaine source est estimée par validation inverse, tandis que l'erreur sur le domaine cible est évaluée par *cross-validation* sur l'échantillon cible automatiquement étiqueté. Notre algorithme stoppe à l'itération l lorsque l'erreur jointe empirique du classifieur inverse, $\hat{err}_S(h_{l+1}^r) + \hat{err}_T(h_{l+1}^r)$,

a atteint un point de convergence ou a augmenté significativement. L'erreur jointe étant positive et bornée par 0, la convergence est assurée. Notons que ce critère est lié à la qualité de la similarité : lorsqu'elle n'est plus suffisamment bonne, l'erreur empirique sur le domaine source augmente dramatiquement et l'algorithme s'arrête. Ce critère permet à la fois le contrôle de la qualité de l'espace de projection construit et de celle de la fonction de similarité.

4.4. L'Algorithme DASF

Notre approche globale, appelée DASF, est présentée dans l'Algo.1. Un point clé de la méthode réside dans le choix des ensembles $U_S \subseteq LS$ et $U_T \subseteq TS$ servant au calcul du couplage biparti $\mathcal{C}_{ST}$. Sachant que l'ensemble des points pertinents permettant une bonne adaptation dépend généralement du problème considéré, nous proposons de les sélectionner à l'aide du classifieur inverse h_l^r et à chaque itération. Ils correspondent aux exemples pour lesquels le classifieur admet une forte ou une faible confiance en ses étiquettes. Soient $\delta_S^+ > 0$, $\delta_T^+ > 0$, $\delta_S^- < 0$, $\delta_T^- < 0$, alors U_S et U_T sont définis par :

$$\left\{ \begin{array}{rcl} U_S & = & \left\{ h_l^r(\mathbf{x}) > \delta_S^+ \text{ OU } h_l^r(\mathbf{x}) < \delta_S^- \mid \mathbf{x} \in LS \right\} \\ U_T & = & \left\{ h_l^r(\mathbf{x}) > \delta_T^+ \text{ OU } h_l^r(\mathbf{x}) < \delta_T^- \mid \mathbf{x} \in TS \right\} \end{array} \right. \text{, tels que } |U_S| = |U_T|.$$

À chaque itération, nous choisissons les paramètres $\delta_{S/T}^{+/-}$ et λ, C de l'Eq.(3) minimisant l'erreur jointe de h_l^r en fonction d'une recherche par grille.

Algorithme 1 DASF

---

entrée Fonction de similarité K, ensemble R, échantillons LS et TS
sortie Classifieur de la dernière itération h_{DASF}

$h_0(\cdot) \leftarrow \sum_{j=1}^{du} K(\cdot, \mathbf{x}'_j)$; $K_1 \leftarrow K$; $l \leftarrow 1$
tant que Le critère d'arrêt n'est pas vérifié **faire**
  Construire $U_S \subseteq LS$, $U_T \subseteq TS$ avec h_{l-1}^r
  Construire $\mathcal{C}_{ST}$
  $\boldsymbol{\alpha}^l \leftarrow$ Résoudre le Pb.(3) avec K_l et $\mathcal{C}_{ST}$
  $K_{l+1} \leftarrow$ MAJ de K_l selon $\boldsymbol{\alpha}^l$
  MAJ R ; $l++$
fin tant que
retourner $h_{DASF}(\cdot) = \sum_{x'_j \in R} \alpha_j^l K_l(\cdot, \mathbf{x}'_j)$

---

5. Expérimentations

Dans cette partie, nous évaluons notre approche DASF sur un problème jouet puis sur une tâche réelle d'annotation d'images. Notre fonction de similarité est basée sur le noyau Gaussien. Pour qu'elle ne soit ni symétrique ni SDP, nous la normalisons telle que chaque similarité vis-à-vis d'un point raisonnable ait une une moyenne de 0 et une variance unitaire sur l'ensemble d'apprentissage. Cependant, cette normalisation dépend des échantillons et n'offre pas toujours la meilleure (ϵ, γ, τ)-qualité par rapport au noyau Gaussien. La similarité avec les meilleures garanties sur le domaine source est choisie *a priori*. Par la suite, les expérimentations faisant appel à une similarité normalisée sont indiquées par *. Comme nous le verrons, elles correspondent généralement aux tâches d'AD plus difficiles. DASF est comparé à un SVM classique, appris uniquement sur le domaine source, et à un *Transductive SVM* semi-supervisé (Vapnik, 1998) (TSVM). Pour ces deux algorithmes, nous utilisons la librairie SVM-light (Joachims, 1999) avec un noyau Gaussien (les paramètres sont choisis par *cross-validation* sur les données sources). De plus, nous nous comparons à un classifieur SF appris uniquement sur le domaine source. La divergence $\hat{d}_{\mathcal{H}}$ entre les deux distributions est estimée à l'aide d'un classifieur SF visant à séparer l'échantillon source du cible. D'après l'Eq.(1), une valeur proche de 0 implique des distributions proches tandis qu'une grande valeur indique une tâche d'AD potentiellement difficile.

5.1. Problème Jouet Synthétique

Ici, notre domaine source correspond à un problème de classification binaire classique de deux lunes jumelles (une classe par lune, cf. Fig.1). Nous considérons 7 domaines cibles différents, chacun produit par une rotation

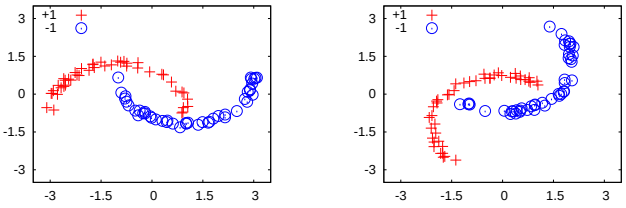

FIG. 1: Gauche : un échantillon source. Droite : un échantillon cible ($50°$).

ANGLE ROTATION	20°	30°	40°	50°	60°*	70°*	80°*	90°*
SVM	89.68	75.99	68.84	60.00	47.18	26.12	19.22	17.2
	±0.78	±0.92	±0.85	±1.08	±2.82	±3.12	±0.28	±0.37
SF	92.4	81.81	72.55	57.85	43.93	39.2	35.93	36.73
	±3.13	±4.62	±7.60	±4.81	±4.46	±9.64	±10.93	±10.17
TSVM	**100**	78.98	74.66	70.91	64.72	21.28	18.92	17.49
	±0.00	±2.31	±2.17	±0.88	±9.10	±1.26	±1.10	±1.12
VS	28	37	37	37	38	35	37	36
	±1.92	±3.77	±2.66	±1.50	±2.67	±2.93	±2.10	±1.69
DASF	99.80	**99.55**	**91.03**	**81.27**	**65.23**	**61.95**	**60.91**	**59.75**
	±0.40	±1.19	±3.30	±4.36	±6.36	±4.88	±2.24	±2.11
RAIS.	**10**	**10**	**9**	**8**	**4**	**4**	**4**	**3**
	±2.32	±1.59	±2.21	±3.27	±0.99	±2.16	±1.84	±1.06
$\hat{d}_{\mathcal{H}}$ DANS ϕ_0^R	0.58	1.16	1.31	1.34	1.34	1.32	1.33	1.31
	±0.04	±0.04	±0.04	±0.04	±0.03	±0.03	±0.03	±0.05
$\hat{d}_{\mathcal{H}}$ DANS ϕ_{final}^R	0.33	0.66	0.82	0.85	0.39	0.40	0.49	0.45
	±0.12	±0.11	±0.13	±0.11	±0.15	±0.05	±0.12	±0.09

TAB. 1: Résultats obtenus pour le problème jouet des lunes jumelles.

anti-horaire (selon 7 angles) du domaine source. Plus l'angle est grand, plus le problème devient difficile. Pour chaque domaine, nous générons 300 instances (150 de chaque classe). La capacité en généralisation des algorithmes est évaluée sur un échantillon de test composé de 1500 exemples tirés selon le domaine cible (et non utilisé par les algorithmes). Chacun des problèmes d'AD est répété 10 fois. La performance moyenne de chaque méthode est calculée et reportée dans la Tab.1. Nous donnons aussi le nombre moyen de Vecteurs Support (VS) calculés par TSVM, le nombre de points raisonnables (RAIS.) trouvés par DASF et une estimation de $\hat{d}_{\mathcal{H}}$ entre les domaines dans les espaces initial ϕ_0^R et final ϕ_{final}^R. Nous faisons les remarques suivantes.

- DASF est en moyenne plus performant. Il est significativement meilleur pour tous les problèmes d'angle supérieur à 20°. Dès 60°, la difficulté augmente et la performance de TSVM chute alors que DASF reste compétitif. La similarité normalisée est alors préférée (*).
- Les points RAIS. sont significativement moins nombreux que les VS. C'est la confirmation que DASF produit des modèles très parcimonieux avec de bonnes performances. Cette quantité est 3 à 12 fois plus faible.
- À la dernière itération, la distance entre les domaines est plus basse. Notre approche rapproche donc bien les distributions. Cependant, l'algorithme tend à construire un "petit" espace pour les problèmes difficiles. Ceci est probablement dû à la nécessité d'avoir des distributions suffisamment proches, et peut induire une perte de performance.

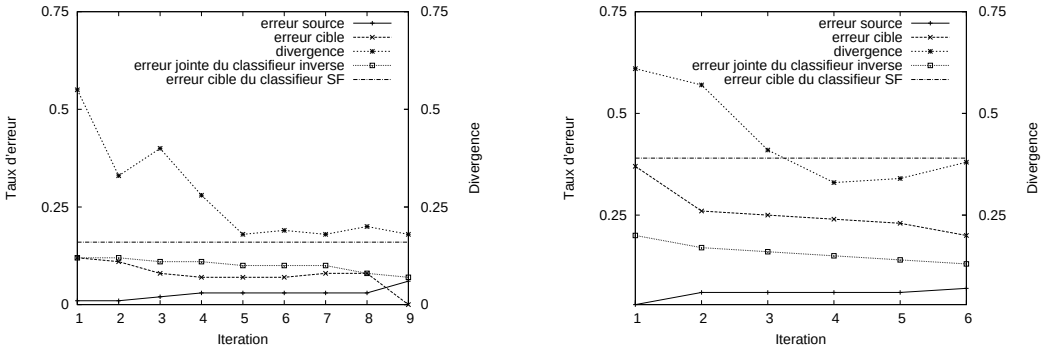

FIG. 2: Deux exécutions de DASF. À gauche pour une rotation de $30°$, à droite de $50°$. En ordonnée, nous lisons à gauche le taux d'erreur, à droite le taux de divergence. Nous observons le taux d'erreur sur les échantillons test source et cible des classifieurs h_i appris à chaque itération, la divergence $\hat{d}_H$ entre les deux distributions, l'erreur jointe du classifieur inverse et l'erreur sur l'échantillon test cible d'un classifieur SF appris sans AD (comme baseline).

La Fig.2 montre l'exécution de DASF sur deux problèmes d'AD de rotations différentes. Dans les deux cas, $\hat{d}_{\mathcal{H}}$ diminue significativement en comparaison à l'itération 1. L'algorithme stoppe quand l'erreur jointe atteint un minimum, après une diminution continue. Pour l'exemple de rotation $30°$, DASF construit un classifieur d'erreur nulle sur l'échantillon de test cible. Pour le problème $50°$, la difficulté est plus grande et DASF infère un classifieur performant, meilleur que le classifieur SF appris seulement sur les données sources. Notons que l'erreur sur l'échantillon source augmente, ce qui est attendu puisque le but est d'être performant sur le domaine cible.

5.2. Classification d'Images

Dans cette partie, nous expérimentons notre approche sur les corpus PascalVOC'07 (Everingham *et al.*, 2007) et TrecVid'07 (Smeaton *et al.*, 2009). L'objectif est l'identification d'objets visuels classiques - de *concepts visuels* - dans des images. Le corpus TrecVid est constitué d'images extraites de vidéos et peut aussi être vu comme corpus d'images. Dans nos expériences,

CONC.	BIRD	BOTTLE*	BUS	CAR	CHAIR	CYCLE	COW
SVM	0.18	0.01	0.16	0.28	0.24	0.10	0.15
VS	867	587	476	1096	1195	392	681
SF	0.17	0.11	0.12	0.33	0.21	0.14	0.11
TSVM	0.14	0.11	0.16	0.37	0.22	0.13	012
SS	814	718	445	631	864	390	888
DASF	**0.19**	**0.12**	**0.17**	**0.38**	**0.26**	**0.16**	**0.16**
RAIS.	**134**	**78**	**94**	**51**	**229**	**192**	**203**

CONC.	DOG*	MONITOR	PERSON*	PLANE	PLANT	SOFA	TRAIN
SVM	**0.24**	**0.16**	0.56	**0.34**	0.12	0.16	0.36
VS	436	698	951	428	428	631	510
SF	0.18	0.12	0.48	0.26	0.13	0.13	0.20
TSVM	0.22	0.12	0.44	0.18	0.10	0.15	0.19
VS	704	861	1111	585	406	866	652
DASF	0.23	0.14	**0.58**	0.33	**0.14**	**0.18**	**0.42**
RAIS.	**42**	**161**	**6**	**141**	**208**	**167**	**75**

	MOY.
SVM	0.22
VS	670
SF	0.19
TSVM	0.19
VS	710
DASF	**0.25**
RAIS.	**127**

TAB. 2: F-mesure sur les domaines cibles PascalVOC. MOY. est le résultat moyen sur les 14 concepts.

nous prenons, comme représentation des images, un descripteur visuel défini par les scores de prédictions sur 15 concepts "intermédiaires" tels que *ciel, herbe, peau,* etc. Chacun d'entre eux est détecté par un classifieur SVM à partir de moments couleurs et d'orientations de contours sur 260 blocs de 32×32 pixels (la dimension est de 3900). Cette représentation a été utilisée avec succès dans de précédentes évaluations TrecVid (Ayache *et al.*, 2007). Nous considérons dans cette section deux expériences.

Tout d'abord, le corpus PascalVOC est constitué d'un ensemble de 5000 images d'apprentissage et 5000 de test. Nous sélectionnons 14 concepts issus de ce corpus et notre but est d'améliorer la performance (en classification) sur l'ensemble de test. Pour ce problème, la divergence entre les ensembles d'apprentissage et de test est faible ($\simeq 0.05$), les deux domaines étant proches. En

FIG. 3: PascalVOC : Les 6 points raisonnables pour le concept PERSON, les 3 premiers sont positifs, les 3 derniers négatifs.

général, pour chaque concept, le ratio d'images positives et négatives (ratio $+/-$) est inférieur à 10%. Nous proposons une évaluation de la capacité d'AD de DASF lorsque ce ratio $+/-$ diffère entre les domaines source et cible, définissant ainsi une tâche d'AD plus difficile. Nous générons un échantillon source par concept, constitué de tous les exemples d'apprentissage positifs et d'exemples négatifs indépendemment tirés tel que le ratio $+/-$ soit $\frac{1}{3}/\frac{2}{3}$. L'échantillon cible est l'échantillon test originel. Pour les 4 méthodes décrites précédemment, nous apprenons un classifieur binaire associé à chaque concept. Comme le ratio $+/-$ est faible sur l'échantillon de test, nous choisissons d'évaluer les performances au sens de la F-mesure et reportons les résultats dans la Tab.2. Remarquons premièrement que TSVM est moins performant, probablement à cause de l'absence supposée d'information sur le domaine cible et donc à l'impossibilité d'estimer le ratio $+/-$. Correctement paramétré, SVM obtient même de meilleures performances que TSVM sur de nombreuses tâches. DASF est, quant à lui, plus performant en moyenne. Il améliore à chaque fois la performance du classifieur SF et est le plus efficace pour 11 concepts. De plus, il construit toujours des modèles significativement plus parcimonieux. En illustration, la Fig.3 est l'ensemble des points RAIS. sélectionnés pour le le concept PERSON.

CONCEPT	CAR*	MONITOR*	PERSON*	PLANE	MOY.
SVM	0.43	0.19	0.52	0.32	0.36
SF	0.52	0.34	0.45	0.54	0.46
TSVM	0.52	**0.37**	0.46	**0.61**	0.49
VS	631	741	1024	259	664
DASF	**0.53**	**0.37**	**0.57**	0.59	**0.52**
RAIS.	**50**	**36**	**19**	**81**	**47**

TAB. 3: F-mesure estimée sur les domaines cibles TrecVid.

Dans la dernière expérience, nous choisissons 4 concepts partagés par TrecVid et PascalVOC. Pour chacun d'entre eux, nous gardons notre domaine source PascalVOC tel quel et prenons, comme domaine cible, un ensemble d'exemples TrecVid de même ratio $+/-$ que le domaine source. $\hat{d}_{\mathcal{H}}$ est de l'ordre de 1.4, les deux corpus sont très différents et la tâche d'AD est donc potentiellement difficile. Les résultats évalués par la F-mesure sont reportés dans la Tab. 3. TSVM et DASF obtiennent des performances similaires, excepté pour le concept PERSON où DASF est significativement meilleur. Nous notons encore une fois que les modèles construits sont à la fois parcimonieux et performants. Enfin, pour ces tâches difficiles la similarité normalisée est souvent préférée (*).

6. Conclusion

Cet article propose une approche d'AD qui tire avantage du cadre de Balcan *et al.* (2008a,b), permettant d'utiliser une fonction de similarité potentiellement non SDP et non symétrique et donc plus flexible. La méthode construit itérativement un espace de projection définie par repondération des similarités vis-à-vis de points raisonnables à la fois proches des exemples sources et cibles. Nous avons montré expérimentalement sur différentes tâches que notre méthode produit des modèles toujours plus parcimonieux et pourvus de bonnes capacités d'adaptation. C'est un avantage certain en perspective d'applications sur des problèmes en grande dimension.

Dans le futur, nous avons l'intention d'étendre notre approche en autorisant l'utilisation de quelques étiquettes cibles, afin d'induire un meilleur espace de projection tel que Daumé III *et al.* (2010). Un autre objectif serait de considérer différents domaines sources (Duan *et al.*, 2009).

Remerciements : Travail financé par le projet ANR VideoSense ANR-09-CORD-026.

Références

AYACHE S., QUÉNOT G. & GENSEL J. (2007). Image and video indexing using networks of operators. *J. Image Video Process.*, **2007**, 1 :1–1 :13.

BALCAN M.-F., BLUM A. & SREBRO N. (2008a). Improved guarantees for learning via similarity functions. In *Proceedings of COLT*, p. 287–298.

BALCAN M.-F., BLUM A. & SREBRO N. (2008b). A theory of learning with similarity functions. *Machine Learning*, **72**(1-2), 89–112.

BEN-DAVID S., BLITZER J., CRAMMER K., KULESZA A., PEREIRA F. & VAUGHAN J. (2010a). A theory of learning from different domains. *Machine Learning Journal*, **79**(1-2), 151–175.

BEN-DAVID S., BLITZER J., CRAMMER K. & PEREIRA F. (2006). Analysis of representations for domain adaptation. In *Proceedings of NIPS'06*.

BEN-DAVID S., LU T., LUU T. & PAL D. (2010b). Impossibility theorems for domain adaptation. *JMLR W&CP*, **9**, 129–136.

BERGAMO A. & TORRESANI L. (2010). Exploiting weakly-labeled web images to improve object classification : a domain adaptation approach. In *Proceedings of NIPS*.

BRUZZONE L. & MARCONCINI M. (2010). Domain adaptation problems : A DASVM classification technique and a circular validation strategy. *IEEE Trans. Pattern Anal. Mach. Intell.*, **32**(5), 770–787.

DAUMÉ III H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the Association for Computational Linguistics (ACL)*.

DAUMÉ III H., KUMAR A. & SAHA A. (2010). Co-regularization based semi-supervised domain adaptation. In *Proceedings of NIPS*.

DUAN L., TSANG I., XU D. & CHUA T. (2009). Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of ICML*. p. 37.

EVERINGHAM M., VAN GOOL L., WILLIAMS C. K. I., WINN J. & ZISSERMAN A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. www.pascal-network.org/challenges/VOC/voc2007/workshop/.

HUANG J., SMOLA A., GRETTON A., BORGWARDT K. & SCHÖLKOPF B. (2006). Correcting sample selection bias by unlabeled data. In *Proceedings of NIPS*, p. 601–608 : MIT Press.

JOACHIMS T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of ICML*, p. 200–209.

MANSOUR Y., MOHRI M. & ROSTAMIZADEH A. (2009). Domain adaptation : Learning bounds and algorithms. In *Proceedings of COLT*, p. 19–30.

PAN S. & YANG Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, **99**(PrePrints).

QUIONERO-CANDELA J., M.SUGIYAMA, SCHWAIGHOFER A. & LAWRENCE N. (2009). *Dataset Shift in Machine Learning*. The MIT Press.

SCHWEIKERT G., WIDMER C., SCHÖLKOPF B. & RÄTSCH G. (2008). An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *Proceedings of NIPS*, p. 1433–1440.

SMEATON A., OVER P. & KRAAIJ W. (2009). High-Level Feature Detection from Video in TRECVid : a 5-Year Retrospective of Achievements. In *Multimedia Content Analysis, Theory and Applications*, p. 151–174. Berlin : Springer Verlag.

SUGIYAMA M., NAKAJIMA S., KASHIMA H., VON BÜNAU P. & KAWANABE M. (2007). Direct importance estimation with model selection and its application to covariate shift adaptation. In *Proceedings of NIPS*.

VAPNIK V. (1998). *Statistical Learning Theory*. Springer.

ZHONG E., FAN W., YANG Q., VERSCHEURE O. & REN J. (2010). Cross validation framework to choose amongst models and datasets for transfer learning. In *Proceedings of ECML-PKDD (Part III)*, p. 547–562 : Springer.